A Nonparametric Procedure to Detect Jumps in Regression Surfaces

Technical Report #626
School of Statistics
University of Minnesota

Peihua Qiu

A NONPARAMETRIC PROCEDURE TO DETECT JUMPS IN REGRESSION SURFACES

Peihua Qiu
School of Statistics
University of Minnesota
270 Vincent Hall
206 Church Street SE
Minneapolis, MN 55455

Abstract

A local smoothing procedure is proposed to detect jump location curves of regression surfaces. In this proposal, jump detection is explicitly related to assumptions on jump location curves. The detected jumps are proved to be strong consistent. This proposal simplifies the computation of some existing local smoothing methods in the statistical literature and weakens their model assumptions as well. It is also connected to the Sobel edge detector in the image processing literature. The problem to evaluate jump detection performance is discussed and a new performance measurement is suggested. Several numerical examples are presented to evaluate the detected jumps and to discuss the selection of the related parameters.

Key Words: Asynchronous window widths; Edge detection; Hausdorff distance; Jump regression surface; Kernel estimation; Performance measurement; Singular points; Sobel edge detector.

1 Introduction

This paper provides a methodology to detect jump locations of regression surfaces. Surface fitting is a fundamental problem in some application fields. For example, meteorologists are interested in fitting the equi-temperature surfaces in high sky or deep ocean. It is many geologists' interest to recover mine surfaces from mineral samples. In many situations, the related surfaces are discontinuous at some places (called jump location curves (JLCs) hereafter). An important example is that image intensity functions have step discontinuities (called *step edges* in image processing) at the outlines of the objects. It is important to detect the JLCs for surface fitting (in some situations

surfaces can be fitted as usual in regions separated by the detected JLCs, see e.g., Müller and Song 1994) and for understanding the surface structure (for example, edges are often regarded as important image structures because much of the image information is conveyed by them, see e.g., Gonzalez and Woods 1992).

In the statistical literature, several proposals have been suggested to detect the JLCs. For example, Korostelev and Tsybakov (1993) suggested using piecewise polynomials for approximating JLCs and the polynomial coefficients were estimated by the maximum likelihood procedure. O'Sullivan and Qian (1994) defined a contrast statistic to detect object boundaries. The boundary curves they considered were the "smooth simple closed" curves. Hall and Raimondo (1997) provided an almost sure convergence rate to approximate a line that separated an entire image into two regions with two different colors. Qiu and Yandell (1997) proposed a jump detection algorithm based on local least squares estimation. Wang (1998) suggested estimating "change curves" via wavelets. Müller and Song (1994) proposed "maximin" estimators of the jump boundaries of the d-dimensional ($d \geq 1$) jump surfaces. Qiu (1997) suggested a so-called *Rotational Difference Kernel Estimator (RDKE)* of the JLCs. The last two methods were both based on two one-sided kernel smoothers along a direction and the estimators were obtained by maximizing the jump detection criteria with respect to this direction. For jump-preserving surface fitting methods, see Chu *et al.* (1998), Qiu (1998) and the references cited there.

Jump detection in regression surfaces is essentially the same problem as edge detection in image processing. Edge detectors (sometimes called filters in image processing) based on gradient estimation are "classic". They make use of the property that estimators of the first order derivatives are large or infinite at edge pixels. Because these methods are intuitive and simple to use, they are included in almost all text books on edge detection (see e.g., Marr and Hildreth 1980; Gonzalez and Woods 1992). More recent edge detection techniques are based on optimal filtering (Canny 1986), random field models (Geman and Geman 1984), surface fitting (Haralick 1984), anisotropic diffusion (Perona and Malik 1990), local smoothing and hypothesis testing (Qiu and Bhandarkar 1996), residual analysis (Chen et al. 1991) and global cost minimization using hill-climbing search (Tan *et al.* 1989), simulated annealing (Tan *et al.* 1991) and the genetic algorithm (Bhandarkar *et al.* 1994).

The above two groups of methods have their own limitations. Most jump detection procedures

in the statistical literature require extensive computation because they are based on numerous maximization/minimization procedures in order to search for the JLCs. For example, the maximization procedures used in some kernel-type methods (Müller and Song 1994; Qiu 1997) need great amount of computation. It is not easy to compute the maximum likelihood estimators of the piecewise polynomial coefficients either in the proposal suggested by Korostelev and Tsybakov (1993). These procedures also impose various model assumptions. Among them a common assumption is that the number of JLCs should be known beforehand, which is hard to be satisfied in real applications. Part of the reason behind these limitations, I think, is that (1) *curves are regarded as and estimated by curves instead of point sets in these methods*; and (2) all directions are considered at each design point for a possible jump direction. Nevertheless, these methods have their own theory to support them, which is important for them to be improved in the future research.

Most edge detection methods in the image processing literature are ready to be used due to the fact that (1) they treat edges as a point set and the detected edges consist of individual pixels; and (2) only a few directions (mostly, the x and y directions) are searched at each pixel. However, some of them do not have enough theory to support them. For example, the Sobel edge detector (see e.g., Rosenfeld and Kak 1982) often uses a 3×3 window (called *mask* in image processing) at each pixel to obtain estimators of the first-order partial derivatives. If the window size increases to $k_1 \times k_2$ with $k_1, k_2 > 3$, how can the Sobel operator be modified accordingly such that the detected edges are statistically consistent (namely, the detected edges converge to the true edges when k's and the image resolution tend to infinity)? Should k_1 and k_2 equal to each other? We have not seen much discussion of this type in the image processing literature yet. Although edge detectors can be evaluated by numerical experiments based on visual impession, we believe that theoretical justifications can help us understand their strengths and limitations such that they can be further improved.

This paper represents part of our research effort to connect (or relate) these two groups of methods. A jump detection procedure is suggested which is based on kernel smoothing techniques. Unlike some existing kernel-type methods, it avoids using the maximization procedure with respect to the direction at each design point, making its computation simple. The number of JLCs can be unknown by this method. Other model assumptions are also flexible. This procedure can be regarded as a generalization of the Sobel edge detector. Its window widths are not restricted to 3×3 any more although they need to satisfy some regularity conditions such that the detected

edges are statistically consistent. Its conditions on the JLCs are explicitly related to the jump detection procedure, which is helpful for users to know the possible places at which this procedure may have difficulty to detect jumps.

At the end of this section, we provide some references on one-dimensional jump regression curve fitting. Some ideas there might be helpful for jump detection in two-dimensional cases although the latter is often more complicated. McDonald and Owen (1986) proposed a "split linear smoother" to fit regression curves with discontinuities preserved. Hall and Titterington (1992) suggested an alternative method by establishing some relations among three local linear smoothers. Müller (1992), Qiu (1994), Qiu et al (1991), Wu and Chu (1993a,b) and Yin (1988), among many others, suggested various kernel-type methods. Eubank and Speckman (1994) treated the jump regression model as a semiparametric regression model and proposed estimators of the jump locations and magnitudes. Loader (1996) suggested a jump detector based on local polynomial kernel estimators. Wang (1995) suggested detecting jumps with wavelet transformations. Qiu and Yandell (1998) developed a jump detection algorithm based on local least squares estimation.

This paper is organized as follows. In Section 2, the jump detection procedure is introduced and its connection to the Sobel edge detector is also discussed. In Section 3, statistical consistency of the detected jumps is established. This procedure is generalized and also related to the existing kernel-type methods in Section 4. A performance measurement of the detected jumps is defined in Section 5. In Section 6, several numerical examples are presented regarding the accuracy of the detected jumps and the selection of the related parameters. Finally, some remarks conclude the article in Section 7.

2 Jump Detection Procedure

Suppose that the regression model concerned is

$$Z_i = f(x_i, y_i) + \varepsilon_i, \quad i = 1, 2, \cdots, n \tag{2.1}$$

where $\{Z_i\}$ are observations, $\{(x_i, y_i)\}$ are design points in design space Ω which is a connected region in R^2, $f(x, y)$ is a bivariate regression function which is continuous in Ω except on some curves (namely, the JLCs), and $\{\varepsilon_i\}$ are i.i.d. errors with mean 0 and variance σ^2.

Throughout this article, we have the following assumption (A 2.1) on the design points:

(A 2.1) There exists a partition $\Lambda := \{\Delta_i, i = 1, 2, \cdots, n\}$ of the design space Ω such that

- $\bigcup_{i=1}^{n} \Delta_i = \Omega$, and $\Delta_i \bigcap \Delta_j = \emptyset$ if $i \neq j$;
- $(x_i, y_i) \in \Delta_i$, for $i = 1, 2, \cdots, n$;
- $\max_{1 \leq i \leq n} d_i = O(n^{-1/2})$, where d_i denotes the diameter of Δ_i;
- $\max_{1 \leq i \leq n} |S(\Delta_i) - 1/n| = O(n^{-1-\lambda})$, where $\lambda > 0$ is some constant and $S(\Delta_i)$ denotes the area of Δ_i.

Remark 2.1 The assumption (A 2.1) requires the design points to have some homogeneity. It is often used in multivariate nonparametric regression analysis (c.f., e.g., Chapter 6, Müller 1988). It is automatically satisfied when the design points form equally spaced squares in Ω.

Let $K_1(x, y)$ and $K_2(x, y)$ be two non-negative kernel functions satisfying: (i) the support of $K_1(x, y)$ is $[-1/2, 1/2] \times [-1, 0]$ and the support of $K_2(x, y)$ is $[-1/2, 1/2] \times [0, 1]$; and (ii) $\int_{-1}^{1} \int_{-1}^{1} K_i(x, y)\, dxdy = 1$ for $i = 1$ and 2. Then we define

$$M_n^{(1)}(x, y) = \frac{1}{nh_n p_n} \sum_{i=1}^{n} Z_i \left[K_2(\frac{x_i - x}{h_n}, \frac{y_i - y}{p_n}) - K_1(\frac{x_i - x}{h_n}, \frac{y_i - y}{p_n}) \right],$$

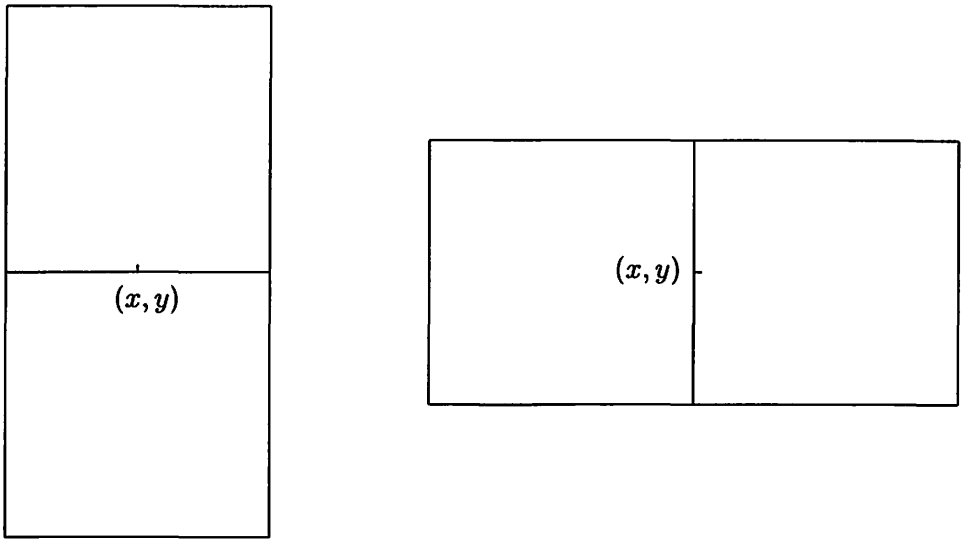
$$M_n^{(2)}(x, y) = \frac{1}{nh_n p_n} \sum_{i=1}^{n} Z_i \left[K_2(\frac{y_i - y}{h_n}, \frac{x_i - x}{p_n}) - K_1(\frac{y_i - y}{h_n}, \frac{x_i - x}{p_n}) \right],$$

and

$$M_n(x, y) = \max \left\{ |M_n^{(1)}(x, y)|, |M_n^{(2)}(x, y)| \right\}, \tag{2.2}$$

where $(x, y) \in \Omega \backslash O(\partial\Omega, b_n)$, $\partial\Omega$ denotes the boundary point set of Ω, $O(\partial\Omega, b_n)$ is the border region of Ω defined by $O(\partial\Omega, b_n) = \{s : d(s, \partial\Omega) \leq b_n \text{ for all } s \in \Omega\}$, $d(\cdot, \cdot)$ is the Euclidean distance, $b_n = \sqrt{h_n^2/4 + p_n^2}$, h_n and p_n are the window widths.

The kernel functions used above are one-sided: $K_1(x, y)$ is lower-sided and $K_2(x, y)$ is upper-sided. The quantity $M_n^{(1)}(x, y)$ is defined as a difference of two weighted averages of the observations in the upper and lower sided neighborhoods of (x, y), respectively (c.f. Figure 2.1 (a)). Similarly $M_n^{(2)}(x, y)$ is a difference of two weighted averages of the observations in the left and right sided neighborhoods of (x, y), respectively (c.f. Figure 2.1 (b)).



(a) (b)

Figure 2.1: (a) The upper and lower sided windows of (x, y) for constructing $M_n^{(1)}(x, y)$; (b) the left and right sided windows of (x, y) for constructing $M_n^{(2)}(x, y)$.

We call (x_0, y_0) a *nonsingular point* of the JLCs if it is on a JLC and satisfies the following three conditions:

(i) The jump magnitude is positive at (x_0, y_0);

(ii) (x_0, y_0) is not a cross point of two or more JLCs;

(iii) Suppose that the JLC, on which (x_0, y_0) is located, has a parametric expression $x = x(t)$ and $y = y(t)$ in a neighborhood of (x_0, y_0), where t denotes the curve length from point (x_0, y_0) to point $(x(t), y(t))$ (obviously, $(x_0, y_0) = (x(t_0), y(t_0))$ with $t_0 = 0$). Then the JLC satisfies the Lipschitz (1) condition at (x_0, y_0) in the sense that: either it can be expressed as $x(t) = \phi(y(t))$ in some neighborhood $N(t_0)$ of t_0 and there exists a constant $M_1 > 0$ such that

$$|x(t_1) - x(t_2)| \leq M_1 |y(t_1) - y(t_2)|, \text{ for any } t_1, t_2 \in N(t_0);$$

or it has an expression $y(t) = \psi(x(t))$ in $N(t_0)$ and there exists another constant $M_2 > 0$ such that

$$|y(t_1) - y(t_2)| \leq M_2 |x(t_1) - x(t_2)|, \text{ for any } t_1, t_2 \in N(t_0),$$

where $\phi(\cdot)$ and $\psi(\cdot)$ are two univariate functions.

If (x_0, y_0) is on a JLC and it is not a nonsingular point, then it is called a *singular point* of the JLCs.

Remark 2.2 The above condition (iii) is essentially the Lipschitz (1) condition which is expressed in a symmetric way with respect to the x and y axes.

Intuitively, $M_n(x, y)$ is close to zero if (x, y) is a continuous point of $f(x, y)$. On the other hand, if (x, y) is a nonsingular point on a JLC, then one of $M_n^{(1)}(x, y)$ and $M_n^{(2)}(x, y)$ is close to the jump magnitude at (x, y). Consequently, $M_n(x, y)$ is relatively large. It is then natural to use point set

$$\hat{D}_n := \{(x_i, y_i) : M_n(x_i, y_i) \geq u_n\}, \tag{2.3}$$

to estimate the point set of the JLCs, $D := \{(x, y) : (x, y)$ is some point on the JLCs$\}$, where u_n is a positive threshold value. In Appendix A, a formula is derived for calculating u_n:

$$u_n = \sqrt{\frac{\sigma^2 \chi^2_{1,\alpha_n/2}}{n h_n p_n} [\int_{-1/2}^{1/2} \int_0^1 (K_2(x, y))^2 \, dxdy + \int_{-1/2}^{1/2} \int_{-1}^0 (K_1(x, y))^2 \, dxdy]}, \tag{2.4}$$

which has the property that when (x, y) is a continuous point and n is large enough,

$$Prob(M_n(x, y) > u_n) \leq \alpha_n,$$

where α_n is a significance level.

We will prove in Section 3 that $\hat{D}_n$ converges to D almost surely in Hausdorff distance under some regularity conditions. One important condition on the window widths is specified by the following (A 2.2). It requires h_n and p_n to be asynchronous. More explanation about (A 2.2) is given in Section 6.2 with a numerical example.

(A 2.2) The window widths h_n and p_n satisfy the condition that $\lim_{n\to\infty} \frac{h_n}{p_n} = 0$.

As mentioned in Section 1, the Sobel operator is a "classic" edge detector in the image processing literature. It is based on two Sobel masks displayed in Figure 2.2. For a given design point (x_i, y_i), a 3×3 neighborhood is considered. Convolution of the first mask with the observations in the neighborhood is used to estimate the partial derivative of the image intensity function with respect to y. This estimator is denoted as $\hat{f}_y^{(i)}$. Similarly, the second mask is used to obtain an estimator of the partial derivative with respect to x, which is denoted as $\hat{f}_x^{(i)}$. Then $[(\hat{f}_x^{(i)})^2 + (\hat{f}_y^{(i)})^2]^{1/2}$ is used as an edge detection criterion with large values indicating possible edges.

1	2	1
0	0	0
-1	-2	-1

-1	0	1
-2	0	2
-1	0	1

Figure 2.2: Sobel masks.

The quantity $\hat{f}_y^{(i)}$ is similar to our $M_n^{(1)}(x_i, y_i)$. It is also a difference of two weighted averages. Similarly, $\hat{f}_x^{(i)}$ is related to $M_n^{(2)}(x_i, y_i)$. The Sobel edge detector used the Euclidean length of the estimated gradient as its edge detection criterion. In (2.2), we suggest using $M_n(x_i, y_i)$ (the maximum of $|M_n^{(1)}(x_i, y_i)|$ and $|M_n^{(2)}(x_i, y_i)|$) to detect jumps based on the following consideration. When (x_i, y_i) is on a JLC, the jump structure of the regression surface contaminates some of the four kernel averages used in constructing $M_n(x_i, y_i)$ (c.f. (2.2)) as estimators of the regression surface at (x_i, y_i). For example, if the JLC is parallel to the x-axis, then the two kernel averages in $M_n^{(1)}(x_i, y_i)$ estimate the surface well. Consequently, $M_n^{(1)}(x_i, y_i)$ is a good estimator of the jump magnitude at (x_i, y_i). But the two kernel averages in $M_n^{(2)}(x_i, y_i)$ do not provide much helpful information for jump detection. It could have negative effect to include them in the jump detection criterion since the criterion would become noisier. This negative effect is mostly eliminated by using $M_n(x_i, y_i)$.

Another important issue is about the window widths. The Sobel edge detector uses windows of size 3×3. With this small size, the detector's ability to smooth out the noise and to enhance the discontinuities is limited. This restriction is lifted in our procedure. In Section 3, some large sample conditions are given on the window widths to guarantee that the detected edges are statistically consistent. From the numerical examples presented in Section 6, we will see that larger window widths do improve the jump detection.

3 Strong Consistency of the Detected Jumps

We establish almost sure consistency of the detected jumps in this section. Since the detected jumps and the true jumps are two point sets in the design space, a measure of distance between two point sets G_1 and G_2 in R^2 must be defined. In this article, the well-known Hausdorff distance

is used, which is defined by

$$d_H(G_1, G_2) := \max\{\sup_{x\in G_1}\inf_{y\in G_2}\|x-y\|, \sup_{x\in G_2}\inf_{y\in G_1}\|x-y\|\}. \tag{3.1}$$

Theorem 3.1 In model (2.1), suppose that the regression function $f(x,y)$ is Lipschitz (1) continuous in the design space Ω except on the JLCs; the design points satisfy the assumption (A 2.1); $E|\varepsilon_1|^p < \infty$ for some $p \geq 2$. Besides the two conditions given in Section 2, $K_1(x,y)$ and $K_2(x,y)$ are assumed to be Lipschitz (1) continuous in their supports. The window widths h_n and p_n are assumed to satisfy the assumption (A 2.2) and the conditions that: (i) $\frac{n^\nu}{\beta_n \log n}[p_n + \frac{1}{n^\lambda h_n p_n} + \frac{1}{\sqrt{n}h_n^2 p_n}] = o(1)$; (ii) $\frac{n^{2\nu}}{nh_n p_n \beta_n} = O(1)$; and (iii) $\frac{n^{\nu+1/p-1}}{\beta_n h_n p_n \log n} = o(1)$, where ν is a positive number and $\{\beta_n\}$ is a series of numbers satisfying $\lim_{n\to\infty}\beta_n = \infty$. The significance level α_n is assumed to satisfy: (i) $\frac{\chi^2_{1,\alpha_n/2}}{nh_n p_n} = o(1)$; and (ii) $\frac{(\beta_n \log n)^2 nh_n p_n}{n^{2\nu}\chi^2_{1,\alpha_n/2}} = O(1)$. Then $\lim_{n\to\infty} d_H(\hat{D}_n \bigcap \Omega_\rho, D \bigcap \Omega_\rho) = 0$, *a.s.*, with rate $O(p_n)$, where $\Omega_\rho := \{s : d(s, \partial\Omega) \geq \rho; d(s, s^*) \geq \rho; s \in \Omega;$ and s^* is some singular point of the JLCs$\}$ and $\rho > 0$ is any constant.

Remark 3.1 By checking the conditions in the above theorem, the convergence rate of $\lim_{n\to\infty} d(\hat{D}_n \bigcap \Omega_\rho, D \bigcap \Omega_\rho) = 0$, *a.s.*, could reach $o(n^{-1/6+\tau})$, for any $0 < \tau < 1/6$. The proof of Theorem 3.1 is given in Appendix B.

4 A Generalization

The jump detection procedure discussed in the previous sections searches the x-axis and y-axis directions for a possible jump at each design point. This idea can be generalized by searching more than two directions as follows. Let $0 \leq \theta_1 \leq \theta_2 \leq \cdots \leq \theta_m < \pi$ be m directions in $[0, \pi)$, where $m \geq 2$ is an integer. At point (x,y), we define

$$M_n^{(i)}(x,y) := \frac{1}{nh_n p_n}\sum_{j=1}^{n} Z_j\left[K_2^{(i)}(\frac{x_j - x}{h_n}, \frac{y_j - y}{p_n}) - K_1^{(i)}(\frac{x_j - x}{h_n}, \frac{y_j - y}{p_n})\right], \quad i = 1, 2, \cdots, m$$

and

$$M_n(x,y) := \max\left\{|M_n^{(i)}(x,y)|, i = 1, 2, \cdots, m\right\}, \tag{4.1}$$

where $K_1^{(i)}(x,y)$ and $K_2^{(i)}(x,y)$ are kernel functions obtained by rotating $K_1(x,y)$ and $K_2(x,y)$ counterclockwisely an angle θ_i. Namely,

$$K_j^{(i)}(x,y) := K_j\left(\delta(x)\sqrt{x^2+y^2}\cos(\arctan(\frac{y}{x}) - \theta_i), \delta(x)\sqrt{x^2+y^2}\sin(\arctan(\frac{y}{x}) - \theta_i)\right), j = 1, 2,$$

where $\delta(x) = 1$ or -1 when $x \geq 0$ or < 0. In applications, we suggest using equally spaced angles in $[0, \pi)$, $\theta_i = \frac{(i-1)\pi}{m}$ for $i = 1, 2, \cdots, m$, unless users have some prior information about jump directions. By using the same arguments as those in Appendix A, the threshold value for $M_n(x, y)$ in (4.1) can be calculated by:

$$u_n = \sqrt{\frac{\sigma^2 \chi^2_{1,\alpha_n/m}}{nh_n p_n} [\int_{-1/2}^{1/2} \int_0^1 (K_2(x,y))^2 \, dxdy + \int_{-1/2}^{1/2} \int_{-1}^0 (K_1(x,y))^2 \, dxdy]}. \tag{4.2}$$

The jump detection criterion (4.1) searches m directions at each design point for a possible jump. By using (4.1), conditions for a point on the JLCs to be nonsingular need to be changed accordingly. More specifically, the condition (iii) in the definition of a nonsingular point given in Section 2 needs to be modified into the following condition (iii)':

(iii)' $\cdots$ Then there exists a neighborhood $N(t_0)$ of t_0 and an angle $\theta \in \{\theta_i\}_{i=1}^m$ such that in this neighborhood the JLC can be expressed as $y(t) = \phi^*(x(t))$, for some Lipschitz (1) continuous function $\phi^*(\cdot)$, after the coordinate system being rotated the angle θ counterclockwisely.

It can be checked that the consistency result in Theorem 3.1 is still true in this case after (2.2) and (2.4) are replaced by (4.1) and (4.2), respectively, and after the condition (iii) in the definition of a nonsingular point is changed to the above (iii)'.

When $m = 2$, $\theta_1 = 0$ and $\theta_2 = \pi/2$, (4.1) is identical to (2.2). On the other hand, when m is large enough such that $\{\theta_i\}_{i=1}^m$ are dense in $[0, \pi)$, the generalized jump detection procedure (4.1)-(4.2) is almost equivalent to the RDKE procedure suggested by Qiu (1997) (also see Müller and Song (1994) for a similar procedure). It is apparent that the value of m is directly related to the amount of computation required and to the conditions on the JLCs as well. If m is chosen larger, then the conditions for a point on the JLCs to be nonsingular are weaker. But the computation involved is more extensive. On the other hand, the computation is simpler when we choose smaller m. But there will be more singular points on the JLCs. In Section 6, a numerical example is presented regarding the selection of m.

5 Performance Measurements of the Detected Jumps

We discuss performance measurement of the detected jumps in this section. Theoretically speaking, the Hausdorff distance (see its definition in Section 3) can be used for this purpose. In reality, this

distance is hard to be calculated. Its computational complexity is $O(n^{3/2})$. Qiu and Yandell (1997) suggested using the so-called averaged performance measurement (APM), which was defined by the averaged distance from the detected jumps to the true JLCs, to measure the performance of jump detection procedures. Some limitations of the APM are obvious. For example, it did not take into account the distance from the true JLCs to the detected jumps.

In this paper, we suggest an alternative performance measurement defined by:

$$d^*(\hat{D}_n, D) := .5\frac{|\hat{D}_n \backslash D|}{|\Omega \backslash D|} + .5\frac{|D \backslash \hat{D}_n|}{|D|} \tag{5.1}$$

where $|A|$ denotes the number of design points in point set A and D can be replaced by $D^* := \{(x_i, y_i) : d((x_i, y_i), D) \leq \sqrt{n}/2\}$ for calculating d^*. In (5.1), $|\hat{D}_n \backslash D|$ is the number of false jump detections and $|\Omega \backslash D|$ is the total number of false jump points in the design space. Therefore $\frac{|\hat{D}_n \backslash D|}{|\Omega \backslash D|}$ is the proportion of the detected false jump points to all false jump points. Similarly, $\frac{|D \backslash \hat{D}_n|}{|D|}$ is the percentage of the true jump points missed by the jump detection procedure. d^* is their average. In two extreme cases that the JLCs are completely detected while there is no false jump detection and that all the false jump points are detected while all the true jumps are missed, d^* equals to 0 and 1, respectively. Generally, d^* is between 0 and 1. The larger its value, the better the performance of the jump detection procedure and vise versa.

Remark 5.1 In (5.1), the two percentages are averaged. In applications, we could also use weighted average: $w\frac{|\hat{D}_n \backslash D|}{|\Omega \backslash D|} + (1-w)\frac{|D \backslash \hat{D}_n|}{|D|}$, where the weight $0 \leq w \leq 1$ represents the relative importance of the first percentage and needs to be specified by users.

Suppose that there is a unique JLC in the design space $[0,1] \times [0,1]$. This JLC is a line parallel to the x-axis at $y = .5$. It is further assumed that there are two sets of detected jump points. The first set consists of the design points on line $y = .5$ and a point $(.8, .1)$ (Figure 5.1(a)). The second set consists of the design points on line $y = .2$ (Figure 5.1(b)). By the Hausdorff distance, the distance between the true JLC and the detected jump points is .4 in the first case and .3 in the case of Figure 5.1(b). Thus it can be concluded that the second set of detected jump points is better than the first set, which might be the opposite to what we would expect. The main reason behind this situation is that the Hausdorff distance is sensitive to individual points (point $(.8, .1)$ in this case). In the definition of d^*, we use proportions instead of supremum/infimum values to make d^* more robust to individual points. It is not hard to check that d^* is close to zero in the case of Figure 5.1(a) and larger than .5 in the case of Figure 5.1(b). Another reason we prefer d^*

is because of its simple computation. It can be checked that its computational complexity is $O(n)$.

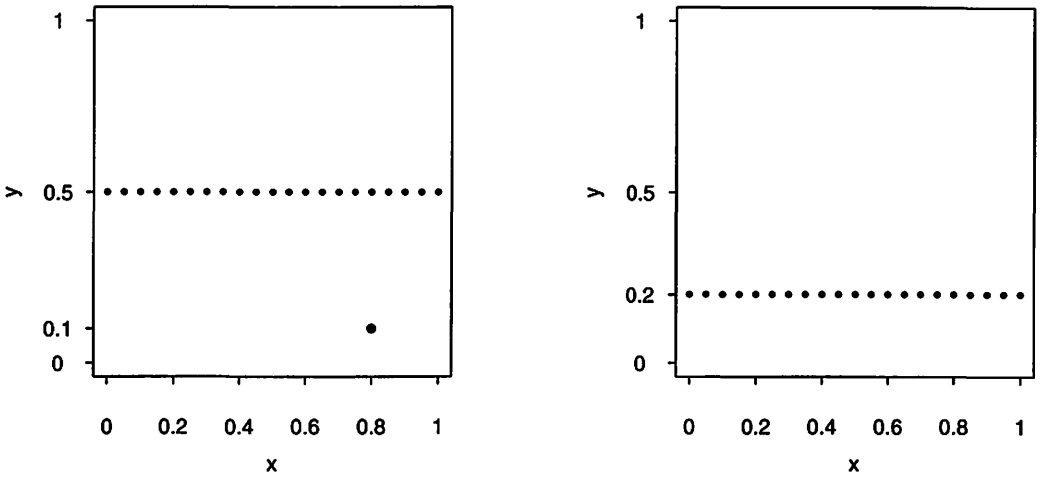


Figure 5.1: Suppose that the true JLC is the line $y = .5$. The set of detected jump points consists of the design points on line $y = .5$ and a point $(.8, .1)$ in plot (a) and of the design points on line $y = .2$ in plot (b).

We should point out that the performance measurement d^* is not perfect for applications. In the example of Figure 5.1, if the detected jumps are the design points on line $y = .5 + 1/n$, then probably we would expect that the jump detection procedure performs better when n gets larger. By d^*, it will not happen. To measure the jump detection performance is an important issue because it is directly related to comparisons of different jump detection procedures. We hope there will be more future research on this topic.

6 Simulation Study

We present our simulation results in two parts. In Section 6.1, numerical performance of the jump detection procedure is evaluated. Then the assumption (A 2.2) is discussed by a numerical example in Section 6.2.

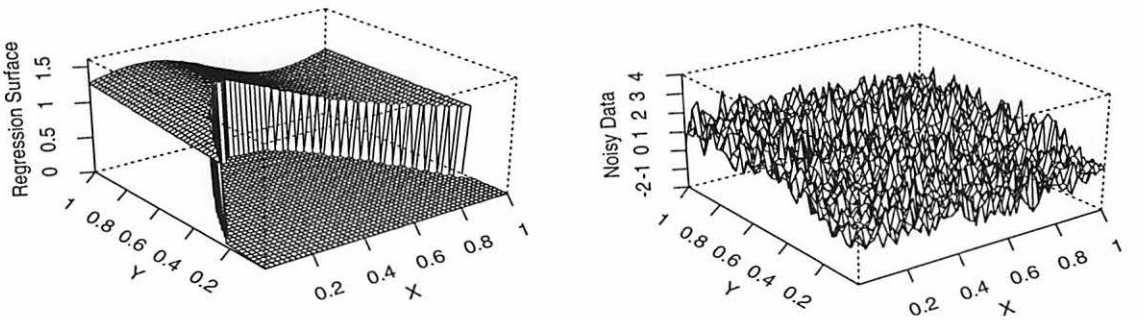
6.1 Numerical Performance of the Jump Detection Procedure

In this part, we present some simulation results to evaluate the performance of the jump detection procedure. The regression function used is $f(x, y) = \frac{1}{4}(1 - x)y + [1 + .2\sin(2\pi x)]I_{y \geq .6\sin(\pi x)+.2}$,

Table 6.1: Performance measurement values (d^*) with several pairs of window widths.

	k_1				
k_2	3	5	7	9	11
3	0.40227	0.28684	0.15510	0.18795	0.16237
5	0.01147	0.01142	0.01723	0.02177	0.02564
7	0.01497	0.02819	0.03683	0.04259	0.04749
9	0.03052	0.04641	0.05718	0.06504	0.07057
11	0.04629	0.06792	0.07792	0.08720	0.08976

for $(x, y) \in [0, 1] \times [0, 1]$, which has a unique JLC $\phi(x) = .6\sin(\pi x) + .2$ with jump magnitude $1 + .2\sin(2\pi x)$. Observations are generated from model (2.1) with $\varepsilon_1 \sim N(0, \sigma^2)$ at design points $(x_i, y_j) = (i/n_1, j/n_1)$, for $i, j = 1, 2, \cdots, n_1$. The sample size is $n = n_1^2$. We define $K_2(x, y) = \frac{12}{11}(1 - x^2) \cdot \frac{12}{11}(1 - (y - .5)^2) I_{[-1/2,1/2]\times[0,1]}$, which is a product of two Epanechnikov kernel functions (see e.g., page 45, Härdle 1991), and $K_1(x, y) = K_2(x, -y)$. The significance level is fixed at $\alpha_n = .01$. For convenience, the window widths h_n and p_n are chosen to be: $h_n = k_1/n_1$ and $p_n = k_2/n_1$, where k_1 and k_2 are two positive odd numbers. Without confusion, sometimes k_1 and k_2 are called the window widths in this section. Figure 6.1 shows the true regression surface (plot (a)) and its noisy version with $\sigma = .5$ (plot (b)).



(a) (b)

Figure 6.1: (a) The true regression surface; (b) a noisy version with $\sigma = .5$.

We first fix $n_1 = 100, \sigma = .5$ and let k_1 and k_2 change. The performance measurement values with several pairs of window widths are summarized in Table 6.1. To eliminate some randomness, all results presented in this section are averages of 100 replications.

As Table 6.1 indicated, the window widths should not be too small or too large. When they are

Table 6.2: For each combination of n_1 and σ, the best pair of window widths (k_1, k_2) and the corresponding d^* value (in parenthesis) are presented.

	n_1			
σ	100	200	300	400
0.25	3,3 (0.00961)	3,3 (0.00469)	3,3 (0.00301)	3,3 (0.00233)
0.50	5,5 (0.01142)	5,5 (0.00564)	5,5 (0.00370)	5,5 (0.00280)
0.75	7,7 (0.01422)	7,7 (0.00701)	7,7 (0.00450)	7,7 (0.00321)
1.0	9,9 (0.01880)	9,9 (0.00784)	9,9 (0.00540)	9,9 (0.00430)

too small, the jump detection criterion $M_n(x, y)$ is still quite noisy, making the threshold value u_n relatively large. Hence some real jump points are missed. On the other hand, when they get larger, more design points have the jump structure involved in their jump detection criterion values. In other words, more design points are detected, making the detected JLC thick (see Figure 6.2(b) and the related discussion given below), which also implies more false jump detections. The best pair of window widths in Table 6.1 is $(k_1, k_2) = (5, 5)$.

Figure 6.2(b) shows the detected jump points with window widths (5,5). We notice that there are two kinds of deceptive jump candidates. The first kind is those scattered in the design space due to the nature of hypothesis testing on which the threshold value is based. The second kind of deceptive jump candidates are those around the true JLC due to the nature of local smoothing. Qiu and Yandell (1997) suggested two modification procedures to delete these two kinds of candidates. Figures 6.2(c) and 6.2(d) show the results after the modification procedures are sequentially applied to the results in plot (b). As a comparison, the true JLC is plotted in Figure 6.2(a).

The above simulation is then repeated with several different values of n_1 and σ. For each combination of n_1 and σ, the best pair of window widths and the corresponding d^* value are presented in Table 6.2.

From Table 6.2, we can see that: (1) d^* gets smaller when n_1 is chosen larger, which may reflect the consistency of the detected jump points; (2) the window widths should be chosen larger if the data is noisier (namely, σ is bigger); (3) the window widths should be quite stable when the sample size changes. Theorem 3.1 tells us that the convergence rate of the detected jumps is $O(p_n)$. Therefore the window widths k_1 and k_2 should be stable to achieve the fastest convergence rate

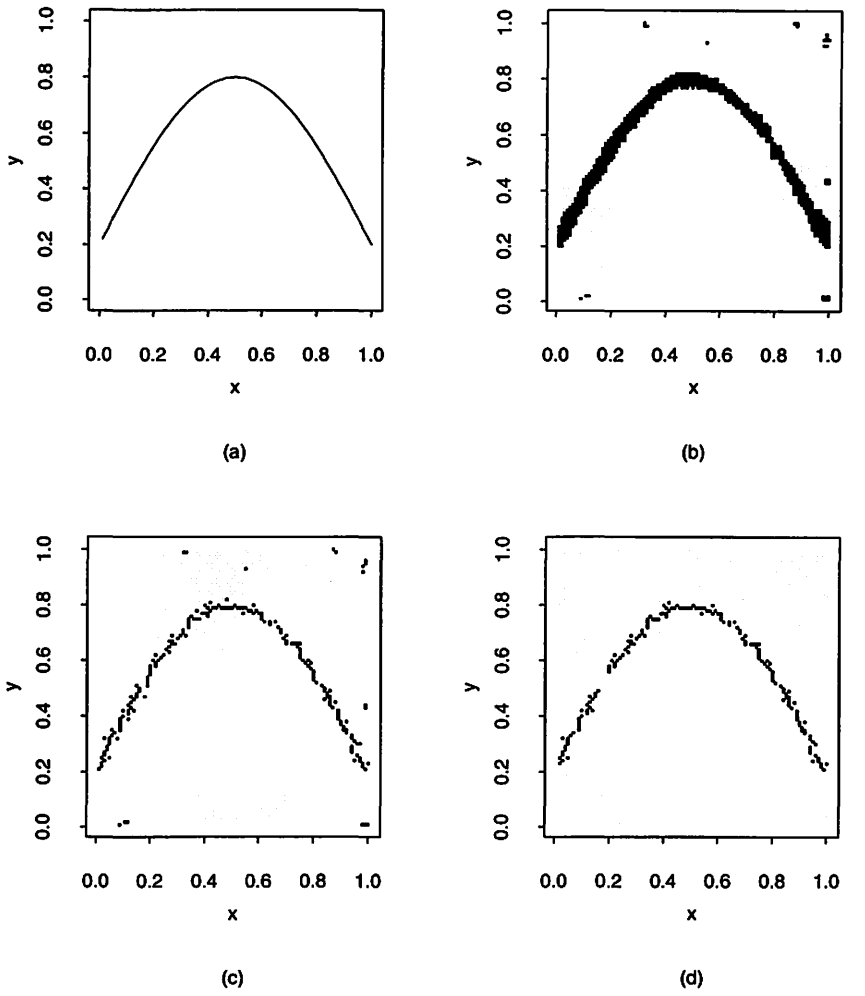


Figure 6.2: (a) The true jump location curve; (b) the detected jump points by the procedure (2.2)-(2.4); (c) the modified version of plot (b) by the first modification procedure in Qiu and Yandell (1997) (to make the detected JLC thinner); (d) the modified version of plot (c) by the second modification procedure in Qiu and Yandell (1997) (to delete some scattered jump candidates).

when the sample size increases. Our simulation results confirm this conclusion. From Table 6.2, it seems that k_1 and k_2 should equal to each other, which conflicts with the assumption (A 2.2). We will explain this issue in Section 6.2.

Next, the generalized jump detection procedure (5.1)-(5.2) is used to detect jumps. Some parameters are chosen to be: $n_1 = 100$, $\sigma = .5$ and $(k_1, k_2) = (5, 5)$. Simulations are performed with 10 different m values: 2, 4, 6, 8, 10, 12, 14, 16, 18 and 20. The amount of computation each set of simulation requires is roughly proportional to the value of m. The performance measurement values are plotted in Figure 6.3. As the plot indicated, it does improve the jump detection to search more directions at each design point. But it spends more computing time at the same time. So

there is a trade-off between the computing time and the accuracy of the detected jumps.

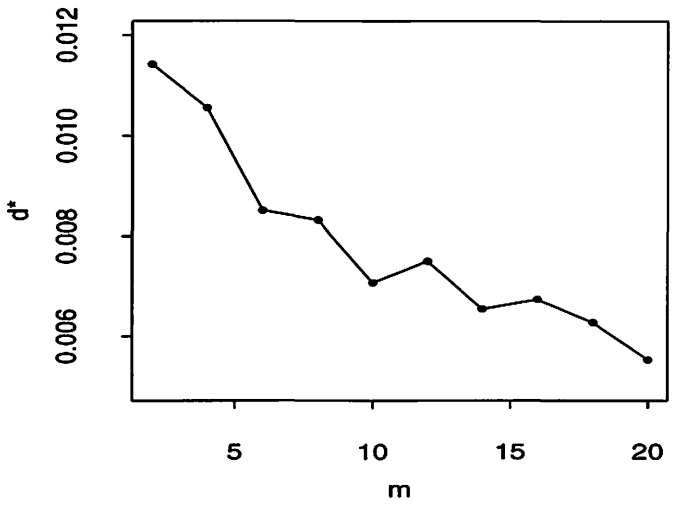


Figure 6.3: Simulation results of the generalized jump detection procedure (5.1)-(5.2).

6.2 Discussion about the Assumption (A 2.2)

From Table 6.2, it seems that the assumption (A 2.2) is not necessary for the procedure to detect jumps efficiently. In this part, we discuss this issue in some detail.

Let us first recall where this assumption is used in Theorem 3.1. From its proof given in Appendix B, this assumption is used in equations (B.6) and (B.7). Without loss of generality, we assume that a JLC has an expression $y = \psi(x)$ in a neighborhood of a given point (x, y) as shown in Figure 6.4 by the dotted curve. The rectangle in the plot represents the support of $K_2(\frac{u-x}{h_n}, \frac{v-y}{p_n})$ as a function of (u, v). Then the ratio of the area of region I (which is below the JLC in the rectangle) to the area of the entire rectangle is of order $O(h_n/p_n)$. (By Lipschitz (1) condition of the JLC, the area of region I is of order $O(h_n^2)$.) The assumption (A 2.2) basically says that the region I is negligible comparing to the entire rectangle. Consequently, $M_n^{(1)}(x, y) \approx C(x, y)$, where $C(x, y)$ is the jump magnitude at (x, y). If $h_n = O(p_n)$ instead, then $M_n^{(1)}(x, y) \approx \gamma C(x, y)$, where γ is a constant between 0 and 1. The value of γ depends on the curvature of the JLC at (x, y) and on the ratio h_n/p_n as well. In the case of Figure 6.1, it can be checked that γ is larger than .5 when n is big enough. Therefore the assumption (A 2.2) is not important in that case. Actually when we know that the JLC is very smooth ("smooth" means small curvature here), then h_n can even be chosen larger than p_n to make the detected JLC thin. However, this kind of window widths are dangerous for detecting jumps at places where the JLC is not smooth enough. This is demonstrated by the

Table 6.3: The best pairs of window widths (k_1, k_2) and the corresponding d^* values (in parentheses) for several combinations of n_1 and c.

	n_1			
c	100	200	300	400
1	3,11 (0.03776)	3,11 (0.01818)	3,11 (0.01141)	3,11 (0.00804)
3	3,11 (0.03827)	3,11 (0.01843)	3,13 (0.01560)	3,13 (0.00992)
5	3,13 (0.05204)	3,13 (0.02601)	3,15 (0.01980)	3,15 (0.01407)
7	3,15 (0.07245)	3,15 (0.03485)	3,15 (0.02013)	3,15 (0.01570)

following example.

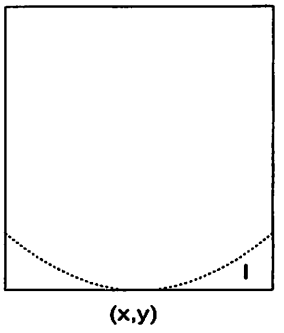


Figure 6.4: The rectangle represents the support of the kernel function $K_2(\frac{u-x}{h_n}, \frac{v-y}{p_n})$ as a function of (u, v). The dotted curve is the JLC on which (x, y) is located. "I" denotes the region below the JLC in the rectangle.

Figure 6.5(a) shows a true JLC which has a sharp angle at point $(.5, .5)$. The JLC has an expression $y = (1 - c)/2 + cx$ when $(c - 1)/(2c) \leq x \leq .5$ and $y = (1 + c)/2 - cx$ when $.5 < x \leq (c + 1)/(2c)$, where $c = 5$. Observations are obtained at regularly spaced design points as we did before and $\sigma = .5$. When $n_1 = 100$ and $(k_1, k_2) = (5, 5)$, a set of detected jump points by the procedure (2.2)-(2.4) is presented in Figure 6.5(b). It can be seen that jump detection around the point $(.5, .5)$ is not good. To further investigate the jump detection at this point, we next concentrate on the cross section of $x = .5$. The performance measurement value is still calculated by formula (5.1), but all quantities in the formula are computed from the design subspace $\{(x, y) : x = .5, 0 \leq y \leq 1\}$. Table 6.3 presents the best pairs of window widths along with the corresponding performance measurement values for several combinations of n_1 and c.

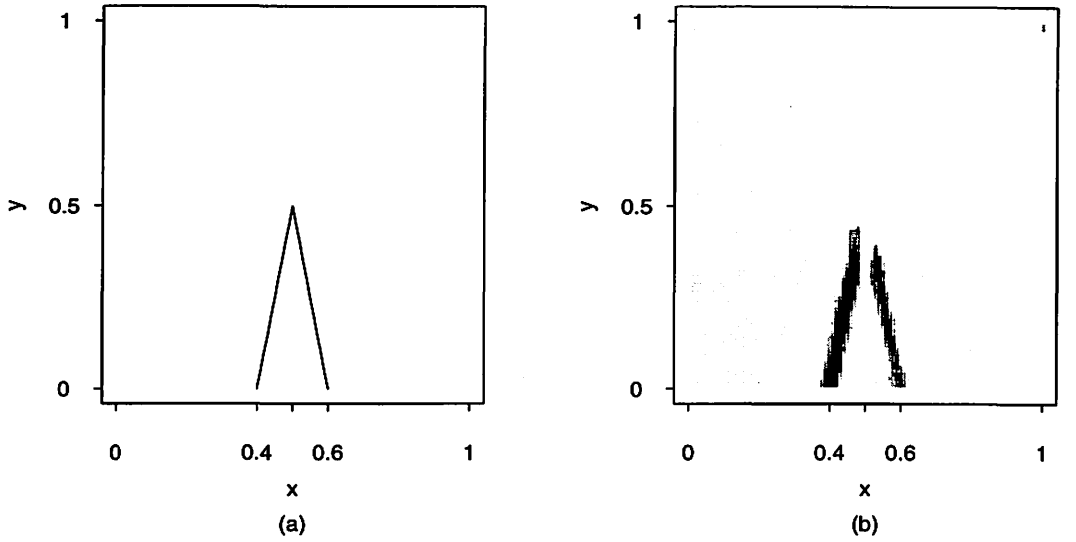


Figure 6.5: (a) The true jump location curve; (b) the detected jump points.

Table 6.3 confirms two points. First, the window widths (k_1, k_2) should be quite stable when n_1 increases as we found in Table 6.2. Second, the ratio k_1/k_2 should be smaller at places where the JLC has larger curvatures (c is larger in this example), which explains the purpose of the assumption (A 2.2). *The assumption (A 2.2) is necessary for the procedure to successfully detect all nonsingular jump points at which the JLCs may have different curvatures.*

7 Concluding Remarks

We have presented a jump detection procedure based on local smoothing techniques. This procedure simplifies the computation of some existing kernel-type methods in the statistical literature and makes their model assumptions more flexible as well. It can also be regarded as a generalization of the Sobel edge detector in image processing. By using our theory, the Sobel edge detector can use masks of size bigger than 3×3 in such a way that the detected edges are statistically consistent.

We are currently concerned about the following issues. First, variable window widths might be more reasonable than the fixed ones. In some planar regions of the surface, we could use large-size windows whereas in highly textured regions, the window sizes could be small. Second, as pointed out in Section 5, the performance measurement d^* still has room for improvement. Third, some parameters in the procedure (such as k_1, k_2 and α_n) have not been well defined yet although some large sample conditions and numerical results are provided. Finally, we have investigated the Sobel

edge detector in some detail. The relationship between the jump detection procedure (2.2)-(2.4) and other edge detectors in the image processing literature needs to be studied in the future.

Appendix

A Derivation of the Threshold Value u_n in (2.4)

First of all,

$$\begin{aligned} & P(M_n(x,y) > u_n) \\ = \; & P(|M_n^{(1)}(x,y)| > u_n \text{ or } |M_n^{(2)}(x,y)| > u_n) \\ \leq \; & P((M_n^{(1)}(x,y))^2 > u_n^2) + P((M_n^{(2)}(x,y))^2 > u_n^2). \end{aligned} \tag{A.1}$$

It is not hard to check that both $M_n^{(1)}(x,y)$ and $M_n^{(2)}(x,y)$ are approximately normally distributed with mean 0 and variance σ_n^2, where

$$\begin{aligned} \sigma_n^2 &= Var(M_n^{(1)}(x,y)) \\ &= \frac{\sigma^2}{(nh_np_n)^2}\sum_{i=1}^{n}[K_2(\frac{x_i-x}{h_n},\frac{y_i-y}{p_n}) - K_1(\frac{x_i-x}{h_n},\frac{y_i-y}{p_n})]^2 \\ &\approx \frac{\sigma^2}{nh_np_n}\int\int_{R^2}[K_2(x,y)-K_1(x,y)]^2\,dxdy \\ &= \frac{\sigma^2}{nh_np_n}[\int_{-1/2}^{1/2}\int_0^1 [K_2(x,y)]^2\,dxdy + \int_{-1/2}^{1/2}\int_{-1}^0 [K_1(x,y)]^2\,dxdy], \end{aligned} \tag{A.2}$$

and " $\approx$ " means that a high order term has been neglected. Therefore if we choose u_n as in (2.4), then by (A.1) and (A.2),

$$P(M_n(x,y) > u_n) \leq P((M_n^{(1)}(x,y))^2/\sigma_n^2 > \chi^2_{1,\alpha_n/2}) + P((M_n^{(2)}(x,y))^2/\sigma_n^2 > \chi^2_{1,\alpha_n/2}) \approx \alpha_n.$$

B Proof of Theorem 3.1

The proof is divided into several parts. In the first part, we will prove that if the regression function $f(x,y)$ is Lipschitz (1) continuous in the design space Ω, then

$$\frac{n^\nu}{\beta_n \log n}\|\hat{f}_n(x,y) - f(x,y)\|_\Omega = o(1), \;\; a.s., \tag{B.1}$$

where $\hat{f}_n(x,y)$ is one of the four kernel estimators involved in the construction of $M_n(x,y)$ and $\|f(x,y)\|_\Omega$ denotes $\max\{|f(x,y)| : (x,y) \in \Omega\}$. Without loss of generality, we assume that $\hat{f}_n(x,y) = \frac{1}{nh_np_n}\sum_{i=1}^n Z_iK_1(\frac{x_i-x}{h_n},\frac{y_i-y}{p_n})$.

To prove (B.1), some arguments from the proof of Theorem 3 in Cheng and Lin (1981) can be borrowed, which proved a one-dimensional version of (B.1). Next, we give an outline of the proof of (B.1) by three steps.

Step 1 By the Lipschitz (1) condition on $f(x,y)$ and $K_1(x,y)$, it is not hard to check that

$$\|E\hat{f}_n(x,y) - f(x,y)\|_\Omega = O(p_n) + O(h_n) + O(\frac{1}{n^\lambda h_n p_n}) + O(\frac{\sqrt{h_n^2+p_n^2}}{\sqrt{n}h_n^2 p_n^2})$$

Step 2 Let

$$\begin{aligned}
\bar{\varepsilon}_i &= \varepsilon_i I_{|\varepsilon_i| \le i^{1/p}}, i = 1, 2, \cdots, n \\
\bar{g}_n(x,y) &= \frac{1}{nh_n p_n}\sum_{i=1}^n \bar{\varepsilon}_i K_1(\frac{x_i - x}{h_n}, \frac{y_i - y}{p_n}) \\
g_n^*(x,y) &= \frac{1}{nh_n p_n}\sum_{i=1}^n \varepsilon_i K_1(\frac{x_i - x}{h_n}, \frac{y_i - y}{p_n}).
\end{aligned}$$

Without loss of generality, we assume that $\Omega \subset [0,1] \times [0,1]$. Let $A_n = \{(i/[n^\eta], j/[n^\zeta]) : i = 1, 2, \cdots, [n^\eta], j = 1, 2, \cdots, [n^\zeta]\} \bigcap \Omega$, where η and ζ are two positive constants and $[x]$ denotes the integral part of x. Then for any $(x,y) \in \Omega$, there exists $(v(x), w(y)) \in A_n$ such that $|x - v(x)| \le 1/[n^\eta]$ and $|y - w(y)| \le 1/[n^\zeta]$. It is obvious that

$$\frac{n^\nu}{\beta_n \log n}\|\bar{g}_n(x,y) - E\bar{g}_n(x,y)\|_\Omega \le S_{1n} + S_{2n} + S_{3n},$$

where

$$\begin{aligned}
S_{1n} &= \frac{n^\nu}{\beta_n \log n}\|\bar{g}_n(x,y) - \bar{g}_n(v(x), w(y))\|_\Omega \\
S_{2n} &= \frac{n^\nu}{\beta_n \log n}\|\bar{g}_n(v(x), w(y)) - E\bar{g}_n(v(x), w(y))\|_\Omega \\
S_{3n} &= \frac{n^\nu}{\beta_n \log n}\|E\bar{g}_n(v(x), w(y)) - E\bar{g}_n(x,y)\|_\Omega.
\end{aligned}$$

By using some similar arguments to those in Cheng and Lin (1981), we can prove that all S_{1n}, S_{2n} and S_{3n} converge to 0 almost surely. Therefore

$$\frac{n^\nu}{\beta_n \log n}\|\bar{g}_n(x,y) - E\bar{g}_n(x,y)\|_\Omega = o(1), \quad a.s.$$

Step 3 Obviously,

$$\begin{aligned}
&\|\hat{f}_n(x,y) - E\hat{f}_n(x,y)\|_\Omega = \|g_n^*(x,y)\|_\Omega \\
\le\ &\|g_n^*(x,y) - \bar{g}_n(x,y)\|_\Omega + \|\bar{g}_n(x,y) - E\bar{g}_n(x,y)\|_\Omega + \|E\bar{g}_n(x,y)\|_\Omega.
\end{aligned}$$

By the fact that $K_1(x,y)$ is a bounded function, it is not difficult to prove that

$$\frac{n^\nu}{\beta_n \log n}\|g_n^*(x,y) - \bar{g}_n(x,y)\|_\Omega = o(1), \quad a.s.,$$

$$\frac{n^\nu}{\beta_n \log n}\|E\bar{g}_n(x,y)\|_\Omega = o(1).$$

The equation (B.1) is proved after combining the above three steps.

Next, we define:

$$O_n(x,y) = \left\{(x',y') : \sqrt{(x'-x)^2 + (y'-y)^2} \le b_n \text{ for } (x',y') \in \Omega \backslash O(\partial\Omega, b_n)\right\},$$

$$D_n = \bigcup_{(x,y)\in D} O_n(x,y).$$

By (B.1), it is obvious that

$$\frac{n^\nu}{\beta_n \log n}\|M_n(x,y)\|_{\Omega\backslash(O(\partial\Omega,b_n)\bigcup D_n)} = o(1), \quad a.s.$$

So when n is large enough, the following expression is true for any $\rho > 0$:

$$\hat{D}_n \bigcap \Omega_\rho \subset D_n \bigcap \Omega_\rho, \quad a.s. \tag{B.2}$$

We now assume that point (x,y) is on a JLC and it is a nonsingular point as well. Without loss of generality, the JLC is assumed to have a function expression $y = \psi(x)$ and $\psi(\cdot)$ satisfies the Lipschitz (1) condition in a neighborhood of (x,y). Clearly,

$$\begin{aligned}
&\frac{1}{nh_np_n}\sum_{i=1}^{n} Z_iK_2(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n}) \\
= &\frac{1}{nh_np_n}(\sum{}' + \sum{}'')Z_iK_2(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n}) \\
= &\frac{1}{nh_np_n}[\sum{}' Z_i + \sum{}''(Z_i + C(x,y))]K_2(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n}) - \\
&\frac{1}{nh_np_n}\sum{}'' C(x,y)K_2(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n})
\end{aligned} \tag{B.3}$$

where $\sum{}'$ denotes summation of the terms which design points are on one side (denoted as side 1) of the JLC and $\sum{}''$ denotes summation of the remaining terms (denoted as side 2). In (B.3), we also assume, without loss of generality, that there is a positive jump from side 2 to side 1 with jump magnitude $C(x,y)$. Similarly,

$$\begin{aligned}
&\frac{1}{nh_np_n}\sum_{i=1}^{n} Z_iK_1(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n}) \\
= &\frac{1}{nh_np_n}[\sum{}'(Z_i - C(x,y)) + \sum{}'' Z_i]K_1(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n}) + \\
&\frac{1}{nh_np_n}\sum{}' C(x,y)K_1(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n})
\end{aligned} \tag{B.4}$$

By (B.1),

$$\begin{aligned}
&\frac{1}{nh_np_n}[\sum{}'Z_i + \sum{}''(Z_i + C(x,y))]K_2(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n}) - \\
&\frac{1}{nh_np_n}[\sum{}'(Z_i - C(x,y)) + \sum{}'' Z_i]K_1(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n}) \\
= \;& C(x,y) + o(\frac{\beta_n \log n}{n^\nu}), \;\; a.s. \qquad \text{(B.5)}
\end{aligned}$$

On the other hand, we can check that the area of the intersection of the support of $K_2(\frac{u-x}{h_n}, \frac{v-y}{p_n})$ with the side 2 of the JLC is at most $M^*(x,y)h_n^2/2$, where $M^*(x,y) > 0$ is a constant, since the JLC is Lipschitz (1) continuous at (x,y). Therefore

$$\begin{aligned}
&\frac{1}{nh_np_n}\sum{}''C(x,y)K_2(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n}) \\
= \;& \frac{M^*(x,y)h_n^2/2}{h_np_n} \cdot \frac{1}{nM^*(x,y)h_n^2/2}\sum{}''C(x,y)K_2(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n}) \\
= \;& O(h_n/p_n) \qquad \text{(B.6)}
\end{aligned}$$

Similarly,

$$\frac{1}{nh_np_n}\sum{}'C(x,y)K_1(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n}) = O(h_n/p_n) \qquad \text{(B.7)}$$

By (B.3)-(B.7), we have

$$M_n^{(1)}(x,y) = C(x,y) + o(\frac{\beta_n \log n}{n^\nu}) + O(h_n/p_n), \;\; a.s.$$

By the fact that $\min_{(x,y)\in D\bigcap\Omega_\rho} M^*(x,y) > 0$ and $\min_{(x,y)\in D\bigcap\Omega_\rho} C(x,y) > 0$, it can be checked that the above equation is uniformly true for $(x,y) \in D\bigcap\Omega_\rho$. So when n is large enough,

$$\hat{D}_n\bigcap\Omega_\rho \supset D\bigcap\Omega_\rho, \;\; a.s. \qquad \text{(B.8)}$$

By (B.2) and (B.8),

$$d_H(\hat{D}_n\bigcap\Omega_\rho, D\bigcap\Omega_\rho) \le d_H(D_n\bigcap\Omega_\rho, D\bigcap\Omega_\rho) \le b_n = \sqrt{h_n^2/4 + p_n^2}, \;\; a.s.,$$

which is the conclusion of the theorem.

References

Bhandarkar, S.M., Zhang, Y., and Potter, W.D. (1994), "An edge detection technique using genetic algorithm-based optimization," *Pattern Recognition 27,* 1159-1180

Canny, J. (1986), "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence 8,* 679-698.

Chen, M.H., Lee, D., and Pavlidis, T. (1991), "Residual analysis for feature detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence 13,* 30-40.

Cheng, K.F., and Lin, P.E. (1981), "Nonparametric estimation of a regression function," *Z. Wahrscheinlichkeitstheorie view. Gebiete 57,* 223-233.

Chu, C.K., Glad, I.K., Godtliebsen, F., and Marron, J.S., (1998), "Edge-preserving smoothers for image processing," *Journal of the American Statistical Association 93,* 526-556.

Eubank, R.L., and Speckman, P.L. (1994), "Nonparametric estimation of functions with jump discontinuities," IMS Lecture Notes, vol.23, *Change-Point Problems* (E. Carlstein, H.G. Müller and D. Siegmund eds.), 130-144.

Geman, S., and Geman, D. (1984), "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images", *IEEE Transactions on Pattern Analysis and Machine Intelligence 6,* 721-741.

Gonzalez, R.C., and Woods, R.E. (1992), *Digital Image Processing,* Addison-Wesley Publishing Company, Inc.

Hall, P., and Raimondo, M. (1997), "Approximating a line thrown at random onto a grid," The Annals of Applied Probability 7, 648-665.

Hall, P., and Titterington, M. (1992), "Edge-preserving and peak-preserving smoothing," *Technometrics 34,* 429-440.

Haralick, R.M. (1984), "Digital step edges from zero crossing of second directional derivatives," *IEEE Transactions on Pattern Analysis and Machine Intelligence 6,* 58-68.

Härdle, W. (1991), *Smoothing Techniques: with implementation in S,* New York: Springer-Verlag.

Korostelev, A.P., and Tsybakov, A.B. (1993), *Minimax Theory of Image Reconstruction*, Lecture Notes in Statistics, Vol. 82, Springer, New York.

Loader, C.R. (1996), "Change point estimation using nonparametric regression," *The Annals of Statistics 24*, 1667-1678.

Marr, D., and Hildreth, E. (1980), "Theory of edge detection," *Proceedings of the Royal Society of London 207*, 187-217.

McDonald, J.A., and Owen, A.B. (1986), "Smoothing with split linear fits," *Technometrics 28*, 195-208.

Müller, H.G. (1988), *Nonparametric Regression Analysis of Longitudinal Data*, Lecture Notes in Statistics, Springer-Verlag: New York.

Müller, H.G. (1992),"Change-points in nonparametric regression analysis," *The Annals of Statistics 20*,737-761.

Müller, H.G., and Song, K.S. (1994), "Maximin estimation of multidimensional boundaries," *Journal of the Multivariate Analysis 50*, 265-281.

O'Sullivan, F., and Qian, M. (1994), "A regularized contrast statistic for object boundary estimation – implementation and statistical evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence 16*, 561-570.

Perona, P., and Malik, J. (1990), "Scale space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 629-639.

Qiu, P. (1994),"Estimation of the number of jumps of the jump regression functions," *Communications in Statistics-Theory and Methods 23*, 2141-2155.

Qiu, P. (1997), "Nonparametric estimation of jump surface," *Sankhyā (Series A) 59*, 268-294.

Qiu, P. (1998), "Discontinuous regression surfaces fitting," *The Annals of Statistics*, (to appear).

Qiu, P., Asano, Chi., and Li, X. (1991), "Estimation of jump regression functions," *Bulletin of Informatics and Cybernetics 24*, 197-212.

Qiu, P., and Bhandarkar, S.M. (1996), "An edge detection technique using local smoothing and statistical hypothesis testing," *Pattern Recognition Letters 17*, 849-872.

Qiu, P., and Yandell, B. (1997), "Jump detection in regression surfaces," *Journal of Computational and Graphical Statistics 6,* 332-354.

Qiu, P., and Yandell, B. (1998), "A local polynomial jump detection algorithm in nonparametric regression," *Technometrics 40,* 141-152.

Rosenfeld, A., and Kak, A.C. (1982), *Digital Picture Processing,* Vols. 1 and 2, Academic Press, New York.

Tan, H.L., Gelfand, S.B., and Delp, E.J. (1989), "A comparative cost function approach to edge detection," *IEEE Transaction on Systems, Man, and Cybernetics 19,* 1337-1349.

Tan, H.L., Gelfand, S.B., and Delp, E.J. (1991), "A cost minimization approach to edge detection using simulated annealing," *IEEE Transactions on Pattern Analysis and Machine Intelligence 14,* 3-18.

Wang, Y. (1995), "Jump and sharp cusp detection by wavelets," *Biometrika 82,* 385-397.

Wang, Y. (1998), "Change curve estimation via wavelets," *Journal of the American Statistical Association 93,* 163-172.

Wu, J.S., and Chu, C.K.(1993a),"Kernel type estimators of jump points and values of a regression function," *The Annals of Statistics 21,* 1545-1566.

Wu, J.S., and Chu, C.K.(1993b),"Nonparametric function estimation and bandwidth selection for discontinuous regression functions," *Statistica Sinica 3,* 557-576.

Yin, Y.Q.(1988),"Detecting of the number, locations and magnitudes of jumps," *Communications in Statistics-Stochastic Models 4,* 445-455.